REMARKS

In summary, claims 1-39 are pending. Claims 1-5, 7-18, 20-26, and 28-38 are rejected under 35 U.S.C. § 102. No claims are amended. No new matter is added.

Telephone Conversation With Examiner

Applicant's representative thanks Examiner Dao for the telephone conversation conducted on April 26, 2007. Examiner Dao and Applicant's representative discussed various claim rejections. In many instances, Applicant's representative explained that he could not find claim limitations in the cited prior art or did not understand how the cited portion of a reference taught a claim limitation. Examiner Dao stated that Applicant's representative should provide such comments in a response to the final rejection.

Claim Rejections - 35 U.S.C. §102

Claims 1-5, 7-18, 20-26, and 28-38 are rejected under 35 U.S.C. §102(a) as being anticipated by an article entitled "Introduction to Series 60 Applications for C++ Developers," Version 1.0, Nokia corporation, August 2002 (hereinafter referred to as "Series 60 for C++").

Applicant's claimed invention is directed to a technique for fast application debugging; thus providing a user of an application development tool the ability to quickly debug an application. Series 60 for C++ does not describe debuggers or how a debugger can provide the ability to quickly debug an application. Series 60 for C++ provides an introduction to creating "Hello World" applications in C++ for Series 60 Platform 1.0. Series 60 for C++ does not provide a description of debuggers.

In accordance with applicant's claimed invention, to achieve fast debugging, specific functions related to debugging are performed in advance of the user invoking the debugger. Prior to launching the debugger: a hosting process, which creates an environment in which the application can be debugged, is started; a runtime environment is loaded in the hosting process; an application domain is created; and the debugger is attached to the hosting process.

Thus, from the user's perspective the time to start debugging is greatly reduced because many of the functions associated with starting the debugger have already been completed when the user invokes the debugger. An exemplary method for debugging an application operating within a hosted runtime environment in accordance with the present invention includes creating a hosting process not based on the application. The hosted runtime environment is created and loaded in a separate process ("hosting process"). The debugger is attached to the hosting process. Then, when a request to debug the application is received, the application is loaded into the hosting process, in response to receiving the request. Series 60 for C++ does not teach the above described technique for fast application debugging.

Per the above mentioned telephone conversation, the following comments are provided.

At page 5 of the instant Office Action, it is asserted that Series 60 for C++ discloses, at page 6, section 3 and page 7 lines 16 and 28-32, "creating a hosting process not based on said application, " as recited in claim 1. The references portions of Series 60 for C++ are copied below.

- Page 6, section 3:

> **3. Symbian OS 6.1**
>
> Central to the success of Series 60 Platform is Symbian OS; it is the foundation of the product.
>
> Symbian OS is a 32-bit multitasking operating system, wherein events often happen asynchronously and applications are designed to interact with one another. For example, a phone call may interrupt a user composing an email message, a user may switch from email to a calendar application in the middle of a telephone conversation, or an incoming SMS may trigger the user to access the contact database and forward the SMS on. By complying with the platform architecture and software design guidelines, application designers can routinely manage such occurrences in the daily lives of smartphone users.

- Page 7, lines 16 and 28-32:

```
\Symbian\6.1\Series60\Epoc32Ex\Basics\HelloWorld
```

> This will build the project for the Series 60 debug emulator. To run the application:
>
> At the command prompt, navigate to the folder that contains the Helloworld.exe application, e.g.,
>
> \Symbian\6.1\Series60\Epoc32\Release\wins\udeb

It is not understood how the above portions of Series 60 for C++ disclose or suggest "creating a hosting process not based on said application, " as recited in claim 1. It is requested that an explanation be provided, in a subsequent Office Action, as to how the above portions of Series 60 for C++ disclose or suggest "creating a hosting process not based on said application."

At page 5 of the instant Office Action, it is asserted that Series 60 for C++ discloses, at page 6, section 2 and page 7 lines 16 and 30, "starting said runtime environment in the hosting process, " as recited in claim 1. The references portions of Series 60 for C++ are copied below.

- Page 6, section 2:

> **2. Series 60 Platform**
>
> Series 60 Platform is a complete smartphone reference design, including a host of wireless applications. The platform builds on the Symbian operating system (Symbian OS), complementing it with a configurable graphical user interface library and a comprehensive suite of reference applications. A set of robust components and many varied APIs are provided in Series 60 Platform. The APIs supplied are used extensively by the suite of "standard" applications but they are designed to be re-used by third-party application developers as well.

- Page 7, lines 16 and 30:

> \Symbian\6.1\Series60\Epoc32Ex\Basics\HelloWorld
>
> \Symbian\6.1\Series60\Epoc32\Release\wins\udeb

It is not understood how the above portions of Series 60 for C++ disclose or suggest "starting said runtime environment in the hosting process, " as recited in claim 1. It is requested that an explanation be provided, in a subsequent Office Action, as to how the above

portions of Series 60 for C++ disclose or suggest "starting said runtime environment in the hosting process."

At page 6 of the instant Office Action, it is asserted that Series 60 for C++ discloses, at page 11, lines 6-11 and page 7 lines 13-16, "attaching a debugger to said hosting process," as recited in claim 1. The references portions of Series 60 for C++ are copied below.

- Page 11, lines 6-11:

> Alternatively, to run the application through the debugger, press F5. Ignore the warning that there is no debug information associated with Epoc.exe. It is the application (essentially a DLL) that will be debugged, not the emulator itself. Check the box so that the warning is not displayed again for this project. The app, and the associated debug information, are located in `Epoc32\Release\wins\udeb\z\system\apps\HelloWorld`.

- Page 7, lines 13-16:

> **5.1.1 Build and Run from the Command Line**
>
> Open a command prompt and change to the drive that contains The Series 60 SDK. Navigate to the folder that contains the project code, e.g., `\Symbian\6.1\Series60\Epoc32Ex\Basics\HelloWorld`

It is not understood how the above portions of Series 60 for C++ disclose or suggest "attaching a debugger to said hosting process, " as recited in claim 1. It is requested that an explanation be provided, in a subsequent Office Action, as to how the above portions of Series 60 for C++ disclose or suggest "attaching a debugger to said hosting process."

At page 6 of the instant Office Action, it is asserted that Series 60 for C++ discloses, at page 11, lines 6-11, "subsequent to said acts of creating, starting, and attaching, receiving a request to debug the application," as recited in claim 1. The references portions of Series 60 for C++ are copied below.

- Page 11, lines 6-11:

> Alternatively, to run the application through the debugger, press F5. Ignore the warning that there is no debug information associated with Epoc.exe. It is the application (essentially a DLL) that will be debugged, not the emulator itself. Check the box so that the warning is not displayed again for this project. The app, and the associated debug information, are located in `Epoc32\Release\wins\udeb\z\system\apps\HelloWorld`.

It is not understood how the above portions of Series 60 for C++ disclose or suggest "subsequent to said acts of creating, starting, and attaching, receiving a request to debug the application, " as recited in claim 1. It is requested that an explanation be provided, in a subsequent Office Action, as to how the above portions of Series 60 for C++ disclose or suggest "subsequent to said acts of creating, starting, and attaching, receiving a request to debug the application."

At page 6 of the instant Office Action, it is asserted that Series 60 for C++ discloses, at pages 10-11, section 5.2.2, Figure 3, and page 7, lines 16 and 28-32 "in response to receiving said request, loading the application into the hosting process," as recited in claim 1. The references portions of Series 60 for C++ are copied below.

- Page 10-11, section 5.2.2:

5.2.2 Build and Run from the IDE

Normally projects such as the HelloWorld application are built and run from within the Microsoft Visual C++ 6.0 IDE as follows:

If the ABLD.BAT file does not already exist (or if the .mmp file or bld.inf file has changed), the build command file must be generated by typing:

```
bldmake bldfiles
```

followed by:

```
abld makefile vc6
```

This will build the project and workspace files (.dsp and .dsw) for Visual C++. They will be located under the `\epoc32\build` sub-folder structure e.g.,

```
\epoc32\build\symbian\6.1\Series60Ex\helloworld\helloworld\wins
```

By opening the workspace file HelloWorld.dsw in Visual C++ the application can be built within the IDE by pressing F7 and then run using Ctrl+F5. When asked for the executable, navigate to Epoc.exe in the folder \Epoc32\Release\wins\udeb in the SDK root. This launches the debug emulator; this is the default for development projects.

Series 60 debug emulator will start up and Series 60 system shell will be shown as in Figure 2. Navigate to the "Other" folder using either the cursor keys on the emulator fascia or the PC cursor keys. Click on the action button in the middle of the cursor controls button to open the folder. Navigate to the HelloWorld application and click on the action button again to invoke the application.

Alternatively, to run the application through the debugger, press F5. Ignore the warning that there is no debug information associated with Epoc.exe. It is the application (essentially a DLL) that will be debugged, not the emulator itself. Check the box so that the warning is not displayed again for this project. The app, and the associated debug information, are located in `Epoc32\Release\wins\udeb\z\system\apps\HelloWorld`.

- Figure 3:

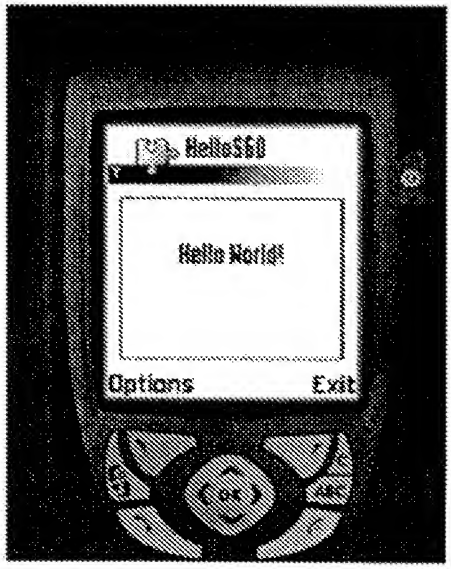

Figure 3: The "Hello World" application

- Page 7, lines 16 and 28-32:

\Symbian\6.1\Series60\Epoc32Ex\Basics\HelloWorld

This will build the project for the Series 60 debug emulator. To run the application:

At the command prompt, navigate to the folder that contains the Helloworld.exe application, e.g.,

\Symbian\6.1\Series60\Epoc32\Release\wins\udeb

It is not understood how the above portions of Series 60 for C++ disclose or suggest "in response to receiving said request, loading the application into the hosting process, " as recited in claim 1. It is requested that an explanation be provided, in a subsequent Office Action, as to how the above portions of Series 60 for C++ disclose or suggest "in response to receiving said request, loading the application into the hosting process."

Because no explanation as to how the cited portions of Series 60 for C++ are being interpreted to rejected the claims, Applicant is denied the opportunity to reasonably rebut the rejections. The arguments provided in the previous Office Action Response (Response dated

February 1, 2007) are maintained. The arguments provided above with respect to independent claim 1 also apply to independent claim 13, independent claim 22, and independent claim 34. Accordingly, it is requested that the rejection, under 35 U.S.C. §102, of claims 1-5, 7-18, 20-26, and 28-38, be reconsidered and withdrawn.

Claims 1, 13, 22, and 34 are rejected under 35 U.S.C. §102(b) as being anticipated by U.S. Patent No. 6,353,923, issued to Bogle *et al.* (hereinafter referred to as "Bogle *et al.*"). Application responded to the rejection of claims 1, 13, 22, and 34 in view of Bogle *et al.* in the previous Office Action Response (Response dated February 1, 2007). These arguments however, have not been addressed in the instant Office Action. Accordingly, it is requested that the response to the rejection of claims 1, 13, 22, and 34 in view of Bogle *et al.* be addressed in a subsequent Office Action, or alternatively, the rejection, under 35 U.S.C. §102, of claims 1, 13, 22, and 34, be reconsidered and withdrawn.

Specifically, Bogle *et al.* neither discloses nor suggests performing specific functions related to debugging in advance of invoking the debugger. Bogle *et al.* teaches an "active debugging environment for debugging a virtual application that contains program language code from multiple compiled and/or interpreted programming languages." (Abstract). The first step of Bogle *et al.*'s debugging process is to activate the debugger, as illustrated in Figure 5 and column 12, lines 43-46: "The active debugging environment operational steps 500 begin at step 508 ..." Thus, active debugging is invoked at the beginning of the flow diagram of Figure 5 of Bogle *et al.*

The arguments provided above with respect to independent claim 1 also apply to independent claim 13, independent claim 22, and independent claim 34. Because Bogle *et al.* neither discloses nor suggests performing specific functions related to debugging in advance of invoking the debugger, and because Bogle *et al.* neither discloses nor suggests "creating a hosting process not based on said application," it is requested that the rejection, under 35 U.S.C. §102, of claims 1, 13, 22, and 34, be reconsidered and withdrawn.

CONCLUSION

In view of the foregoing arguments, remarks, and amendments, it is respectfully submitted that this application is in condition for allowance. Reconsideration of this application and an early Notice of Allowance are respectfully requested. In the event that the Examiner cannot allow this application for any reason, the Examiner is encouraged to contact the undersigned attorney to discuss resolution of any remaining issues.

Date: May 30, 2007

/Joseph F. Oriti/
Joseph F. Oriti
Registration No. 47,835

Woodcock Washburn LLP
Cira Centre
2929 Arch Street, 12th Floor
Philadelphia, PA 19104-2891
Telephone: (215) 568-3100
Facsimile: (215) 568-3439